CHAPTER 23

Process-to-Process Delivery:

Solutions to Review Questions and Exercises

Review Questions

1. ***Reliability*** is not of primary importance in applications such as echo, daytime, BOOTP, TFTP and SNMP. In custom software, reliability can be built into the client/server applications to provide a more reliable, low overhead service.
2. IP and UDP are both ***connectionless*** and ***unreliable protocols***. The main difference in their reliability is that IP only calculates a checksum for the IP header and not for the data while UDP calculates a checksum for the entire datagram.
3. ***Port addresses*** do not need to be universally unique as long as each IP address/port address pair uniquely identify a particular process running on a particular host. A good example would be a network consisting of 50 hosts, each running echo server software. Each server uses the well known port number 7, but the IP address, together with the port number of 7, uniquely identify a particular server program on a particular host. Port addresses are ***shorter*** than IP addresses because their domain, a single system, is smaller than the domain of IP addresses, all systems on the Internet.
4. ***Ephemeral*** is defined as short-lived or transitory. Ephemeral port numbers are only used for the duration of a single communication between client and server, so they are indeed short-lived.
5. The minimum size of a UDP datagram is **8** bytes at the transport layer and **28** bytes at the IP layer. This size datagram would contain no data–only an IP header with no options and a UDP header. The implementation may require padding.
6. Since the length of a datagram must be contained in a 2 byte field, the maximum size of a UDP datagram is **65,535** bytes (header plus data). However, given that the IP layer must also store the total length of the packet in a 2 byte field, the maximum length would be 20 bytes less than this, or **65,515** bytes, to leave room for the IP header. The implementation may impose a smaller limit than this.
7. The smallest amount of process data that can be encapsulated in a UDP datagram is **0** bytes.

8. The largest amount of process data that can be encapsulated in a UDP datagram is **65,507** bytes. (65,535 minus 8 bytes for the UDP header minus 20 bytes for the IP header). The implementation may impose a smaller limit than this.
9. See Table 23.1.

Table 23.1 *Answer to the Question 9.*

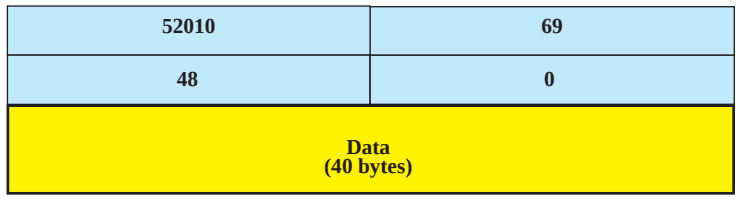
Fields in UDP	*Fields in TCP*	*Explanation*
Source Port Address	**Source Port Address**	
Destination Port Address	**Destination Port Address**	
Total Length		**There is no need for total length.**
Checksum	**Checksum**	
	Sequence Number	**UDP has no flow and error control.**
	Acknowledge Number	**UDP has no flow and error control.**
	Header Length	**UDP has no flow and error control.**
	Reserved	**UDP has no flow and error control.**
	Control	**UDP has no flow and error control.**
	Window Size	**UDP has no flow and error control.**
	Urgent Pointer	**UDP cannot handle urgent data.**
	Options and Padding	**UDP uses no options.**

10. ***UDP*** is preferred because each user datagram can be used for each chunk of data. However, a better solution is ***SCTP***.
11.
    a. None of the control bits are set. The segment is part of a data transmission without piggybacked acknowledgment.
    b. The ***FIN*** bit is set. This is a FIN segment request to terminate the connection.
    c. The ***ACK*** and the ***FIN*** bits are set. This is a ***FIN+ACK*** in response to a received ***FIN*** segment.
12. The ***maximum size*** of the TCP header is **60** bytes. The ***minimum size*** of the TCP header is **20** bytes.

Exercises

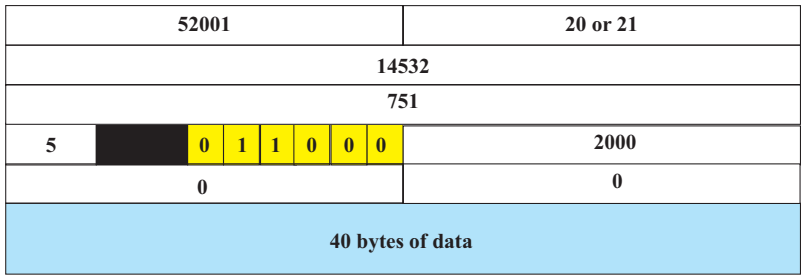
13. See Figure 23.1.
14. The client would use the IP address **122.45.12.7**, combined with an ephemeral port number, for its source socket address and the IP address **200.112.45.90**, combined with the well- known port number **161**, as the destination socket address.
15. The server would use the IP address **130.45.12.7**, combined with the well-known port number **69** for its source socket address and the IP address **14.90.90.33**, combined with an ephemeral port number as the destination socket address.
16. This datagram ***cannot be transferred*** using a single user datagram.

Figure 23.1 *Solution to Exercise 13*



17. 16 bytes of data / 24 bytes of total length = **0.666**
18. 16 bytes of data / 44 bytes of total length = **0.364**
19. 16 bytes of data / 72 byte minimum frame size = **0.222**
20.
    a. Port number **1586**
    b. Port number **13**
    c. **28** bytes
    d. **20** bytes (28 – 8 byte header)
    e. ***From a client to a server***
    f. ***Daytime***
21. It looks as if both the destination IP address and the destination port number are corrupted. ***TCP calculates the checksum and drops the segment***.
22. 0111 in decimal is 7. The total length of the header is 7 × 4 or **28**. The base header is **20** bytes. The segment has **8** bytes of options.
23. See Figure 23.2.

Figure 23.2 *Solution to Exercise 23*



24.
    a. The source port number is **0x0532** (**1330** in decimal).
    b. The destination port number is **0x0017** (**23** in decimal).
    c. The sequence number is **0x00000001** (**1** in decimal).

d. The acknowledgment number is **0x00000000** (**0** in decimal).

e. The header length is **0x5** (**5** in decimal). There are 5 × 4 or **20** bytes of header.

f. The control field is **0x002**. This indicates a SYN segment used for connection establishment.

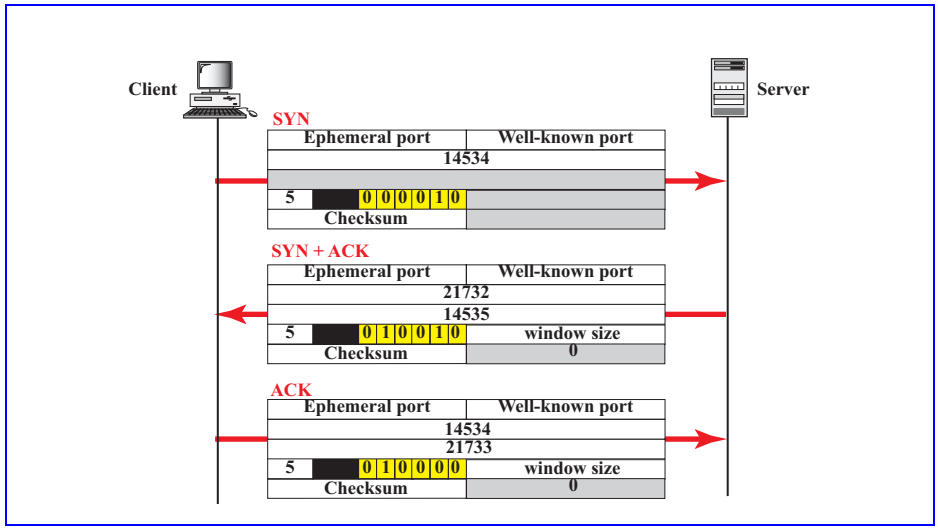
g. The window size field is **0x07FF** (**2047** in decimal).

25. Every second the counter is incremented by 64,000 × 2 = **128,000**. The sequence number field is 32 bits long and can hold only 2^{32}–1. So it takes (2^{32}–1)/(128,000) seconds or **33,554** seconds.

26. 3000 – 2000 = **1000** bytes

27. See Figure 23.3.

Figure 23.3 *Solution to Exercise 27*



28.

a. **At TCP level:**

16 bytes of data / (16 bytes of data + 20 bytes of TCP header) ≈ **0.44** or **44** percent

b. **At IP level**

16 bytes of data / (16 bytes of data + 20 bytes of TCP header + 20 bytes of IP header) ≈ **0.29** or **29** percent

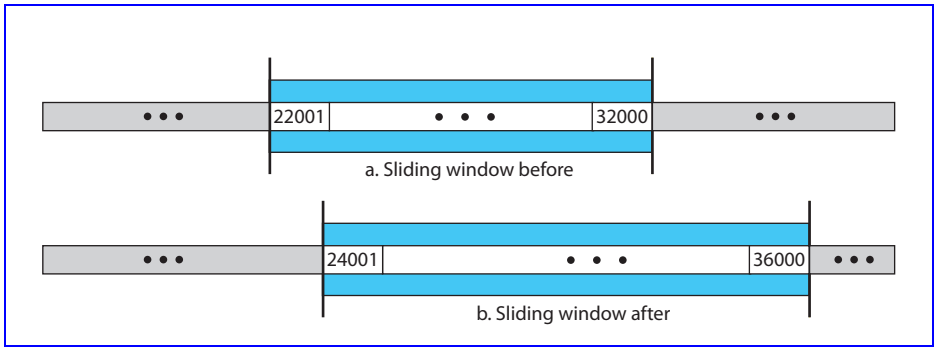
c. **At data link level (**assuming no preamble or flag**):**

16 bytes of data / (16 bytes of data + 20 bytes of TCP header + 20 bytes of IP header + 18 bytes of Ethernet header and trailer ≈ **0.22** or **22** percent,

29. The largest number in the sequence number field is 2^{32} –1. If we start at 7000, it takes [(2^{32} – 1) –7000] / 1,000,000 = **4295** seconds.

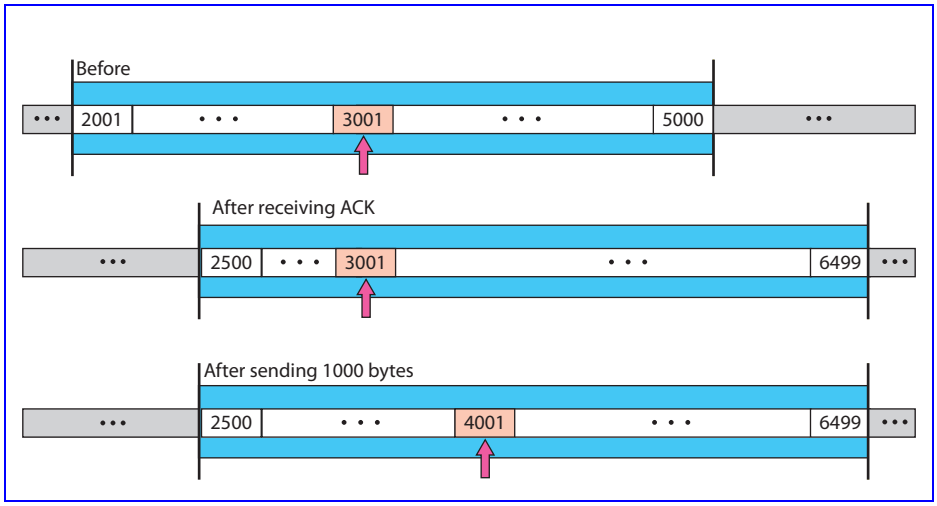
30. See Figure 23.4.

Figure 23.4 *Solution to Exercise 30*



31. See Figure 23.5.

Figure 23.5 *Solution to Exercise 31*



32. The SACK chunk with a cumTSN of **23** was delayed.
33. See Figure 23.6.
    Note that the value of cumTSN must be updated to 8.

34. See Figure 23.7. Chunks 18 and 19 are sent but not acknowledged (200 bytes of data). 18 DATA chunks (1800 bytes) can be sent, but only 4 chunks are in the queue. Chunk **20** is the next chunk to be sent.

Figure 23.6 *Solution to Exercise 33*

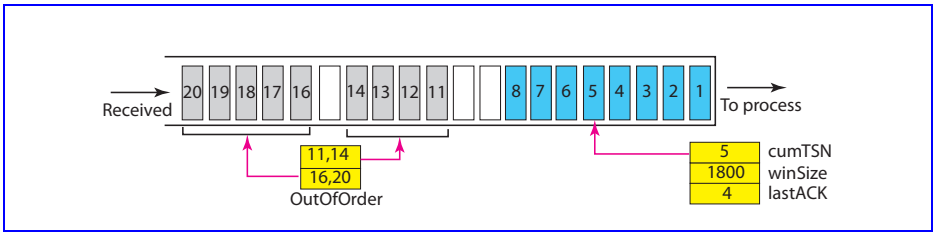


Figure 23.7 *Solution to Exercise 34*